

Санкт-Петербургский государственный университет

*ПОВАЛЯЕВ Никита Денисович*

Выпускная квалификационная работа

*Разработка программно-аппаратного комплекса  
для управления системой роботов*

Уровень образования: бакалавриат

Направление 02.03.02 «Фундаментальная информатика и  
информационные технологии»

Основная образовательная программа СВ.5003.2016 «Программирование  
и информационные технологии»

Профиль «Автоматизация научных исследований»

Научный руководитель:

доцент, кафедра механики управляемого движения,  
к.ф.-м.н. Шиманчук Дмитрий Викторович

Рецензент:

главный специалист, Главное управление по организации работы с  
персоналом СПбГУ, к.ф.-м.н. Лепихин Тимур Андреевич

Санкт-Петербург

2020 г.

# Содержание

<b>Введение</b> . . . . .	3
<b>Постановка задачи</b> . . . . .	5
<b>Обзор литературы</b> . . . . .	6
<b>Глава 1. Математические модели</b> . . . . .	7
1.1. Прямая задача о положении . . . . .	9
1.2. Обратная задача о положении . . . . .	10
1.3. Выбор ближайшего решения . . . . .	17
1.4. Перевод координат . . . . .	17
1.5. Модель колесных роботов . . . . .	17
1.6. Определение координат объектов . . . . .	18
<b>Глава 2. Архитектура системы</b> . . . . .	19
<b>Глава 3. Программная реализация</b> . . . . .	21
3.1. Общее описание . . . . .	21
3.2. Формат команд для управления . . . . .	23
3.3. Интерфейсы для управления . . . . .	26
3.4. Интерфейс построения сложных команд . . . . .	28
3.5. Система технического зрения . . . . .	29
<b>Заключение</b> . . . . .	30
<b>Список литературы</b> . . . . .	31
<b>Приложение</b> . . . . .	32

## Введение

Роботы позволяют выполнять операции, которые недоступны человеку по различным причинам, таким как физическая сложность, монотонность, опасные условия или же действия, связанные с работой в космосе. Но роботы несовершенны и сами по себе очень ограничены. Им необходимы «органы чувств» в виде разнообразных датчиков для восприятия информации об окружающей среде для взаимодействия с объектами и система управления. Существует множество подходов к реализации такой системы: от задания последовательности команд до разработки полноценного искусственного интеллекта.

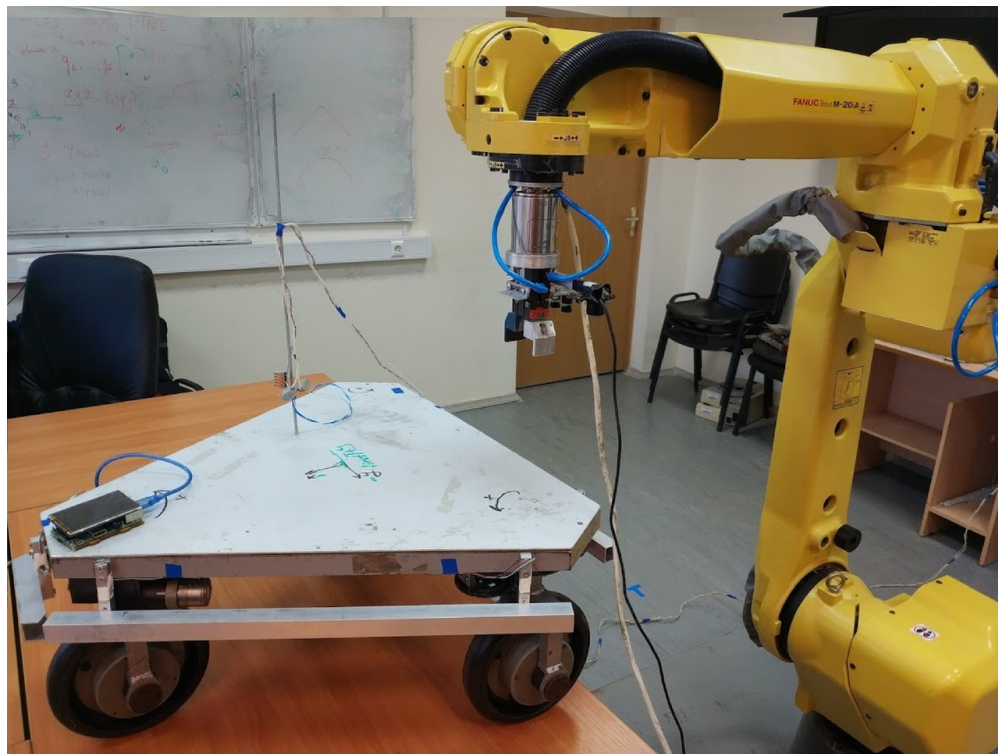
В качестве одного из таких подходов можно выделить систему дистанционного управления, состоящую из компьютерной модели сцены, обновляющейся динамически при помощи использования показаний датчиков, и предоставляющую интерфейсы для различных режимов управления роботами. Такими режимами могут быть:

- выполнение требуемых операций в режиме симуляции без использования реальных роботов и окружения, что позволит сократить риски и затраты для проверки различных гипотез,
- копирующее управление,
- задание последовательностей высокоуровневых команд посредством специального интерфейса,
- сохранение последовательности команд по выполненным в симуляции операциям, для их дальнейшего воспроизведения при необходимости.

В данной работе будут рассмотрены реализации первых трех пунктов.

Для построения прототипа такой системы и полноценной проверки ее работоспособности необходимо иметь несколько видов роботов, оборудованных различными датчиками, чтобы проверить универсальность предложенного и реализованного подхода для возможного дальнейшего расширения системы на большее число роботов.

Конкретные роботы, для которых велась разработка и на которых проводились эксперименты, представлены на рис. 1.



Трехколесный мобильный робот

Модернизированный манипулятор  
Fanuc M-20iA

**Рис. 1:** Роботы

Fanuc M-20iA оборудован шестиосевым силомоментным датчиком, пневматическим схватом и камерой. Трехколесный робот несет на себе плату Raspberry PI для приема команд и передачи управления на каждое из колес, приводимых в движение сервоприводом, и также оборудован датчиками по периметру для детектирования столкновений с объектами.

## Постановка задачи

Разработать прототип комплекса для обеспечения различных режимов управления системой роботов разных типов. Для этого необходимо решить ряд задач:

- описать кинематические модели используемых роботов,
- реализовать кинематические модели на языке высокого уровня. Для роботов-манипуляторов решения прямой и обратной задачи на основе параметров Денавита-Хартенберга будет лучше представить в общем виде, чтобы для использования в системе других роботов необходимо было приложить минимум усилий: задать сами параметры и несколько функций. Для колесных роботов тоже лучше всего задать обобщенную и простую модель, но это не отменяет возможности при необходимости реализовать нечто более сложное и точное,
- создать 3D-модели используемых роботов и окружения,
- создать пользовательский интерфейс для управления роботами,
- предложить архитектурный подход, поддерживающий работу с несколькими типами роботов,
- реализовать режим копирующего управления и режим выполнения последовательности действий.

## Обзор литературы

В [1] была приведена математическая модель трехколесного робота, но для поставленных в рамках серии экспериментов задач она не подходила, и было принято решение использовать другую, более простую реализацию. Трехмерная модель использована та же, что и в этой работе.

В [2] и [3] приведено исчерпывающее описание математических моделей разнообразных роботов. Отсюда была взята основная часть формул для решения обратной задачи кинематики для шестизвенного робота.

В [4] был предложен подход к копирующему управлению при помощи специальной рукояти. В данной работе предлагается упростить управление при помощи компьютерных моделей, а не физического устройства. Также в ней были описания силомоментного датчика, алгоритма его калибровки и управления роботом с его помощью, что можно интегрировать с системой симуляции для отображения воздействия внешних сил на рабочий инструмент для визуализации взаимодействия с объектами окружения.

В [5] было описана и реализована кинематика робота Fanuc, но это было сделано на языке MATLAB, что не годилось для интеграции с другими прикладными программами. Также в MATLAB есть встроенные методы для решения алгебраических и матричных уравнений, чего нет по умолчанию в других языках. И параметры Денавита-Хартенберга, представленные в [5], не совпадали с параметрами используемого в экспериментах робота.

## Глава 1. Математические модели

Рассмотрим кинематическую модель шестизвенных роботов манипуляторов на основе метода Денавита-Хартенберга на примере робота Fanuc M-20ia, схема которого представлена на рис. 2.

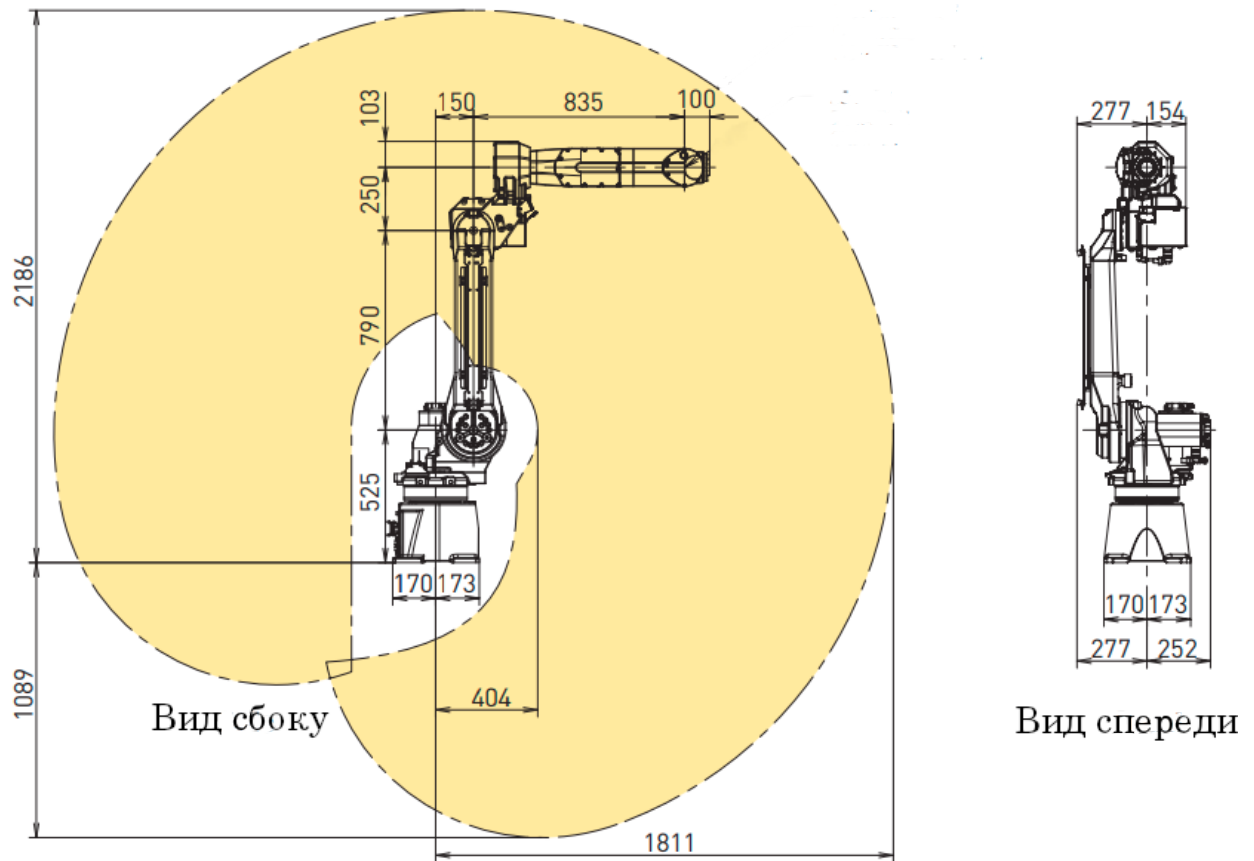


Рис. 2: Схема робота

### Представление Денавита-Хартенберга

Суть заключается в построении по определенному алгоритму систем координат, связанных с каждым звеном манипулятора, что позволяет использовать матрицы перехода между текущей и предыдущей системами координат и дает возможность последовательно преобразовать координаты рабочего инструмента, связанного с последним звеном, в базовую систему координат.

Алгоритм построения систем координат и определения параметров:

1. Построение абсолютной системы координат.

Построить правую ортогональную систему координат  $O_0X_0Y_0Z_0$ , направив  $Z_0$  вдоль оси первого сочленения в направлении схвата.

2. Инициализация и цикл.

Для всех  $i = 1, 2, \dots, N$  выполнить шаги 3-6.

3. Построение  $Z_i$ . Направить ось  $Z_i$  вдоль оси  $(i + 1)$ -го шарнира. При  $i = N$  (т.е. для схвата) выберем ось  $Z_N$  в направлении оси  $Z_{N-1}$ .

4. Построение начала  $i$ -й системы координат.

Выбрать начало  $i$ -й системы координат в точке пересечения осей  $Z_{i-1}$  и  $Z_i$  или в точке пересечения оси  $Z_i$  и общей нормали к осям  $Z_{i-1}$  и  $Z_i$ , если оси  $Z_{i-1}$  и  $Z_i$  не пересекаются.

5. Построение оси  $X_i$ .

Направить ось  $X_i$  вдоль общей нормали к осям  $Z_{i-1}$  и  $Z_i$ .

6. Построение оси  $Y_i$ .

Направить ось  $Y_i$  так, чтобы полученная в результате система  $O_iX_iY_iZ_i$  была правосторонней.

7. Нахождение параметров.

Для всех  $i = 1, 2, \dots, N$  выполнить шаги 8-11.

8. Нахождение  $d_i$ .

Параметр  $d_i$  равен расстоянию от начала  $(i - 1)$ -й системы координат до точки пересечения осей  $Z_{i-1}$  и  $X_i$ , измеренному в направлении оси  $Z_{i-1}$ . Если  $i$ -е сочленение телескопическое, то  $d_i$  является обобщенной координатой.

9. Нахождение  $a_i$ .

Параметр  $a_i$  равен расстоянию от точки пересечения осей  $Z_{i-1}$  и  $X_i$  до начала  $i$ -й системы координат, измеренному в направлении оси  $X_i$ .



10. Нахождение  $q_i$ .

Параметр  $q_i$  равен углу поворота оси  $X_{i-1}$  вокруг оси  $Z_{i-1}$  до ее совпадения с осью  $X_i$ . Если  $i$ -е сочленение вращательное, то  $q_i$  является обобщенной координатой.

11. Нахождение  $\alpha_i$ .

Параметр  $\alpha_i$  равен углу поворота оси  $Z_{i-1}$  вокруг оси  $X_i$  до ее совпадения с осью  $Z_i$ .

12. Конец.

Для используемого в экспериментах робота были получены следующие значения параметров, представленные в табл. 1. Параметры  $q'_i$  здесь обозначают обобщенные координаты, отображающиеся на пульте управления роботом.

Звено	$q$ , рад	$d$ , мм	$a$ , мм	$\alpha$ , рад
1	$q'_1$	0	150	$\pi/2$
2	$-q'_2 + \pi/2$	0	790	0
3	$q'_2 + q'_3$	0	250	$\pi/2$
4	$-q'_4$	835	0	$-\pi/2$
5	$q'_5$	0	0	$\pi/2$
6	$-q'_6$	100	0	0

**Таблица 1:** Таблица параметров

Таблица с ограничениями углов поворота звеньев:

	Звено 1	Звено 2	Звено 3	Звено 4	Звено 5	Звено 6
min	-170°	-70°	-70°	-200°	-140°	-270°
max	170°	90°	200°	200°	140°	270°

**Таблица 2:** Таблица ограничений

## 1.1 Прямая задача о положении

Суть прямой задачи состоит в нахождении положения и ориентации рабочего инструмента робота в абсолютной системе координат по вектору обобщенных координат (углов поворота каждого из звеньев).

Так как все параметры Денавита-Хартенберга известны, можно построить матрицы перехода между  $i$ -ой и  $(i-1)$ -ой системами координат (1), перемножив матрицу перехода, осуществляющую сдвиг на  $d_i$ , с матрицей перехода, осуществляющей поворот на  $q_i$ , с матрицей перехода, осуществляющей сдвиг на  $a_i$ , и с матрицей перехода, осуществляющей поворот на  $\alpha_i$ . Оси сдвигов и переходов соответствуют смысловому значению параметров.

$$T_i^{i-1} = \begin{pmatrix} \cos q_i & -\cos \alpha_i \sin q_i & \sin \alpha_i \sin q_i & a_i \cos q_i \\ \sin q_i & \cos \alpha_i \cos q_i & -\sin \alpha_i \cos q_i & a_i \sin q_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1)$$

Для нахождения координат шестого звена относительно абсолютной системы координат используется следующее выражение:

$$T = T_1^0 T_2^1 T_3^2 T_4^3 T_5^4 T_6^5.$$

## 1.2 Обратная задача о положении

Суть обратной задачи состоит в нахождении вектора обобщенных координат по известному положению последнего звена в исходной системе координат.

### Задача позиционирования

Координаты робота в абсолютной декартовой системе заданы следующим вектором:  $x, y, z$  — положение шестого звена,  $w, p, r$  — углы Эйлера, описывающие поворот шестого звена:

$$coord = (x, y, z, w, p, r). \quad (2)$$

Прежде всего необходимо найти центр сферического запястья: пусть  $C = (x_c, y_c, z_c)$  — пересечение осей 4, 5 и 6, так называемый центр сферического запястья. Положение точки  $C$  зависит только от значений углов

$q_1, q_2, q_3$  и не зависит от значений углов  $q_4, q_5$  и  $q_6$ .

Для нахождения этой точки можно воспользоваться следующей интерпретацией матрицы ориентации шестого звена (геометрический смысл представлен на рис. 3):

$$T = \begin{pmatrix} n_x & \tau_x & \beta_x & p_x \\ n_y & \tau_y & \beta_y & p_y \\ n_z & \tau_z & \beta_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

где

- $n$  — вектор нормали к схвату (перпендикулярен пальцам схвата),
- $\tau$  — касательный вектор схвата (лежит в плоскости пальцев схвата),
- $\beta$  — вектор подхода схвата,
- $p$  — вектор положения схвата.

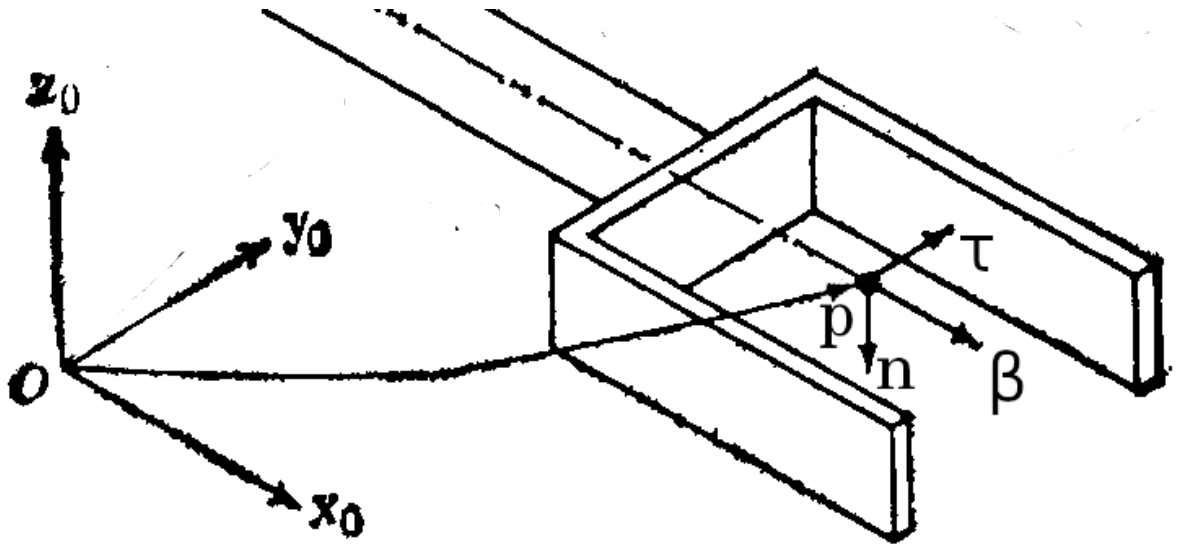


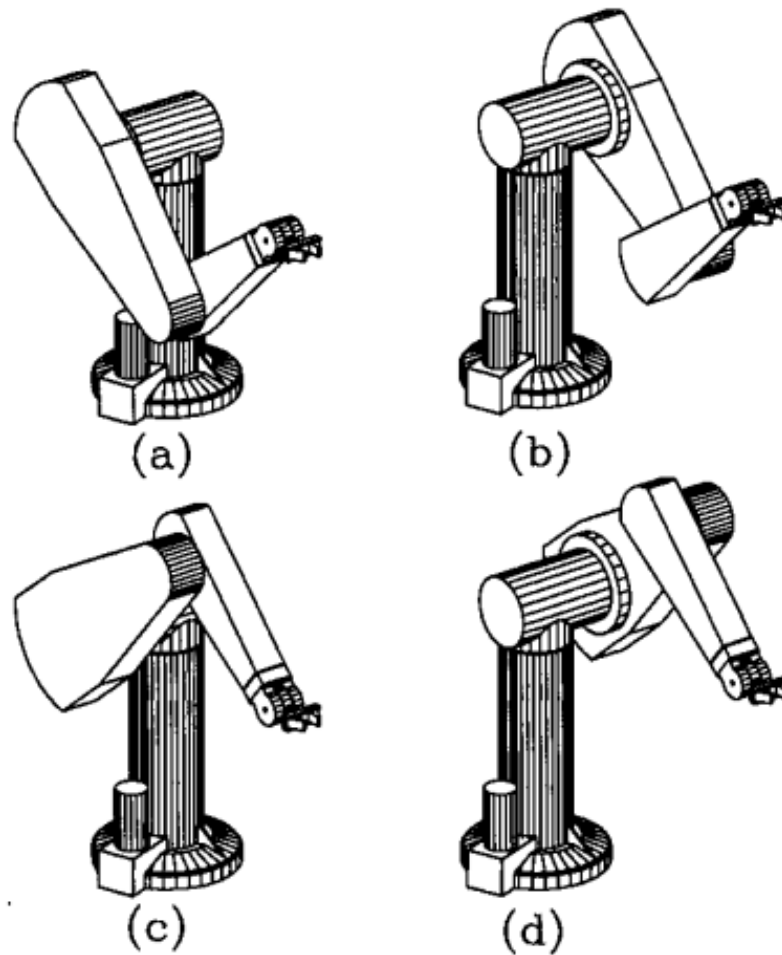
Рис. 3: Геометрическая интерпретация

Для того чтобы вычислить координаты точки  $C$ , нужно вычислить матрицу поворота из углов  $w, p$  и  $r$ , взять третий столбец  $\beta$ , соответствующий вектору подхода схвата, и для точки  $C$  получится выражение:

$$C = \begin{pmatrix} x & y & z \end{pmatrix} - d_6 \begin{pmatrix} \beta_x & \beta_y & \beta_z \end{pmatrix},$$

где  $d_6$  — соответствующий параметр Денавита-Хартенберга.

Цель задачи позиционирования состоит в том, чтобы найти углы первых трех звеньев соответствующих координатам точки  $C$ . Таких вариантов положения может быть несколько, как показано на рис. 4.



**Рис. 4:** Варианты решения задачи о положении

На следующем шаге нужно вычислить следующие вспомогательные коэффициенты  $A - V$ , подробный вывод которых приведен в [2]:

$$A = 2a_1x_c,$$

$$B = 2a_1y_c,$$

$$C = 2a_2a_3 - 2d_2d_4 \sin \alpha_2 \sin \alpha_3,$$

$$D = 2a_3d_2 \sin \alpha_2 + 2a_2d_4 \sin \alpha_3,$$

$$E = a_2^2 + a_3^2 + d_2^2 + d_3^2 + d_4^2 - a_1^2 - x_c^2 - y_c^2 - (z_c - d_1)^2 + \\ 2d_2d_3 \cos \alpha_2 + 2d_2d_4 \cos \alpha_2 \cos \alpha_3 + 2d_3d_4 \cos \alpha_3,$$

$$F = y_c \sin \alpha_1,$$

$$G = -x_c \sin \alpha_1,$$

$$H = -d_4 \sin \alpha_2 \sin \alpha_3,$$

$$I = a_3 \sin \alpha_2,$$

$$J = d_2 + d_3 \cos \alpha_2 + d_4 \cos \alpha_2 \cos \alpha_3 - (z_c - d_1) \cos \alpha_1,$$

$$R = 4a_1^2(J - H)^2 + \sin^2 \alpha_1(E - C)^2 - 4a_1^2 \sin^2 \alpha_1(x_c^2 + y_c^2),$$

$$S = 4(4a_1^2I(J - H) + \sin^2 \alpha_1D(E - C)),$$

$$T = 2(4a_1^2(J^2 - H^2 + 2I^2) + \sin^2 \alpha_1(E^2 - C^2 + 2D^2) - 4a_1^2 \sin^2 \alpha_1(x_c^2 + y_c^2)),$$

$$U = 4(4a_1^2 I(J + H) + \sin \alpha_1^2 D(E + C)),$$

$$V = 4a_1^2(H + J)^2 + \sin^2 \alpha_1(E + C)^2 - 4a_1^2 \sin^2 \alpha_1(x_c^2 + y_c^2).$$

Далее нужно составить уравнение:

$$R\tau_3^4 + S\tau_3^3 + T\tau_3^2 + U\tau_3 + V = 0.$$

Пусть его корнями будут  $(\tau_3)_i, i = 1, 2, 3, 4$ , тогда возможные варианты для  $q_3$  будут выглядеть следующим образом:

$$(q_3)_i = 2 \arctan (\tau_3)_i, i = 1, 2, 3, 4.$$

Далее на основе найденных  $q_3$  вычисляются  $\sin q_1$  и  $\cos q_1$  по формулам:

$$\cos q_1 = \frac{-G(C \cos q_3 + D \sin q_3 + E) + B(H \cos q_3 + I \sin q_3 + J)}{\Delta_1},$$

$$\sin q_1 = \frac{F(C \cos q_3 + D \sin q_3 + E) - A(H \cos q_3 + I \sin q_3 + J)}{\Delta_1},$$

где  $\Delta_1 = AG - FB$ .

Само значение  $q_1$  вычисляется с использованием функции *atan2*.

Для каждого  $q_1$  вычисляются аналогично  $\sin q_2$  и  $\cos q_2$  по формулам:

$$\cos q_2 = \frac{A_{11}(x_c \cos q_1 + y_c \sin q_1 - a_1) - A_{12}(-x_c \cos \alpha_1 \sin q_1 + y_c \cos \alpha_1 \cos q_1 + (z_c - d_1) \sin \alpha_1)}{\Delta_2},$$

$$\sin q_2 = \frac{A_{12}(x_c \cos q_1 + y_c \sin q_1 - a_1) + A_{11}(-x_c \cos \alpha_1 \sin q_1 + y_c \cos \alpha_1 \cos q_1 + (z_c - d_1) \sin \alpha_1)}{\Delta_2},$$

где

$$A_{11} = a_2 + a_3 \cos q_3 + d_4 \sin \alpha_3 \sin q_3,$$

$$A_{12} = -a_3 \cos \alpha_2 \sin q_3 + d_3 \sin \alpha_2 + d_4 \sin \alpha_3 \cos \alpha_2 \cos q_3 + \\ d_4 \sin \alpha_2 \cos \alpha_3,$$

$$\Delta_2 = A_{11}^2 + A_{12}^2.$$

После получения значений углов  $q_1, q_2, q_3$  можно произвести обратное преобразование к преобразованию углов  $q$ , указанному в табл. 1, и отбросить те варианты, которые не подходят по ограничениям, чтобы сократить объемы вычислений.

## Задача ориентации

Из заданных в (2) углов  $w, p, r$  нужно посчитать матрицу поворота  $R$ :

$$R = R_x(w)R_y(p)R_z(r).$$

С другой стороны, ее можно представить следующим образом:

$$R = R_{03}R_{36},$$

где  $R_{03}$  — матрица поворота третьего звена,  $R_{36}$  — матрица поворота шестого звена относительно третьего.  $R_{36}$  может быть найдена как

$$R_{36} = R_{03}^{-1}R.$$

Обозначим компоненты третьего столбца матрицы  $R_{36}$ , показывающие вектор подхода схвата относительно системы координат третьего звена:

$$e_{36} = \begin{pmatrix} \xi \\ \eta \\ \zeta \end{pmatrix}.$$

Аналогично для матрицы  $R$  возьмем вектор  $e$  и для матрицы  $R_{03}$  — вектор  $e_{03}$ . Если  $e$  и  $e_{03}$  коллинеарны, значит, робот оказался в точке сингулярности, и дальнейшее решение обратной задачи невозможно, если не коллинеарны, то решение продолжается.

Значение  $\tau_4 = \tan(q_4/2)$  можно найти следующим образом:

$$\tau_4 = \frac{\xi \sin \alpha_4 \pm \sqrt{(\xi^2 + \eta^2) \sin^2 \alpha_4 - (\cos \alpha_5 - \zeta \cos \alpha_4)^2}}{\cos \alpha_5 - \zeta \cos \alpha_4 - \eta \sin \alpha_4},$$

что дает два возможных значения  $q_4$ .

Для каждого  $q_4$  можно посчитать значения синуса и косинуса для углов  $q_5$  и  $q_6$  по следующим формулам:

$$\sin q_5 = \frac{(\sin \alpha_6 r_{12} + \cos \alpha_6 r_{13}) \cos q_4 + (\sin \alpha_6 r_{22} + \cos \alpha_6 r_{23}) \sin q_4}{\sin \alpha_5},$$

$$\cos q_5 = \frac{-\cos \alpha_4 (\sin \alpha_6 r_{12} + \cos \alpha_6 r_{13}) \sin q_4 + \cos \alpha_4 (\sin \alpha_6 r_{22} + \cos \alpha_6 r_{23}) \cos q_4 + \sin \alpha_4 (\sin \alpha_6 r_{32} + \cos \alpha_6 r_{33})}{-\sin \alpha_5},$$

$$\cos q_6 = (r_{11} \cos q_4 + r_{21} \sin q_4) \cos q_5 + (-\cos \alpha_4 (r_{11} \sin q_4 - r_{21} \cos q_4) + \sin \alpha_4 r_{31}) \sin q_5,$$

$$\sin q_6 = -\cos \alpha_5 (r_{11} \cos q_4 + r_{21} \sin q_4) \sin q_5 + \cos \alpha_5 (-\cos \alpha_4 (r_{11} \sin q_4 - r_{21} \cos q_4) + \sin \alpha_4 r_{31}) \cos q_5 + (\sin \alpha_4 (r_{11} \sin q_4 - r_{21} \cos q_4) + \cos \alpha_4 r_{31}) \sin \alpha_5.$$

Таким образом, чтобы получить конечный результат, нужно вычислить значения самих углов при помощи функции *atan2* и применить преобразование, обратное к преобразованию углов  $q$ , указанному в табл. 1. После этого необходимо провести финальную проверку на ограничения и отбросить неподходящие значения.



### 1.3 Выбор ближайшего решения

Для обратной задачи о положении возможны несколько корректных решений и предполагается использование режима управления в реальном времени, когда неизвестна траектория движения и следующая точка может быть задана посредством устройства ввода любого типа, например, клавиатуры, эмулятора пульта управления или силомоментного датчика. Требуется из всех возможных решений выбрать такой вектор обобщенных координат, который будет иметь минимальное абсолютное суммарное изменение по каждой обобщенной координате относительно тех координат, в которых робот находится в текущий момент времени. Формально это можно записать как

$$Q_{nearest} = \operatorname{argmin}_{Q_i \in Q} \left( \sum_{j=1}^n |Q_i[j] - Q_{current}[j]| \right),$$

где  $Q$  — множество решений,  $n$  — число обобщенных координат.

### 1.4 Перевод координат

Система координат робота отличается от системы координат, заданной в сцене Unity3D. Они имеют разную ориентацию и разные единицы измерения по осям, но при этом обе являются неизменяемыми во времени, поэтому можно один раз посчитать константные матрицы перехода и использовать их всегда, когда нужно перевести координаты реального мира в координаты смоделированной сцены. Это может понадобиться при любых видах действий с объектами: отобразить в 3D-сцене найденный при помощи камеры объект или по координатам объекта в сцене вычислить, в какую точку перевести робота, чтобы его взять, и так далее.

### 1.5 Модель колесных роботов

Существует множество способов описания кинематической модели колесных роботов, но, с учетом известных сценариев использования в рамках запланированных экспериментов, было принято решение выбрать са-

мую простую модель — модель материальной точки, в которой текущее положение задается вектором из двух координат  $(x, y)$ , а следующая точка для перемещения — вектором  $(x', y')$ , и робот движется по прямой от текущей точки к следующей, предварительно повернувшись в нужном направлении всем корпусом или же развернув только колеса.

## 1.6 Определение координат объектов

Робот-манипулятор оборудован камерой. Предположим, что некоторым образом можно получить координаты объектов на сцене относительно камеры в виде матрицы перехода  $T_{obj}^c$  из системы координат камеры в систему координат объекта. Далее предположим, что камера имеет фиксированную позицию относительно шестого звена робота и можно измерить все сдвиги и повороты ее системы координат относительно системы координат шестого звена. Тогда можно посчитать матрицу перехода из системы координат шестого звена в систему координат камеры  $T_c^6$ . Пусть известны обобщенные или декартовы координаты робота, тогда можно найти матрицу, описывающую положение шестого звена  $T$ . В случае обобщенных координат для этого нужно решить прямую задачу, в случае задания положения схвата манипулятора в декартовых координатах в абсолютном пространстве — преобразовать к матричному виду. Тогда можно посчитать координаты объекта в системе координат робота следующим образом:

$$T_{obj} = TT_c^6T_{obj}^c.$$

## Глава 2. Архитектура системы

Высокоуровневая архитектура комплекса для управления системой роботов представлена на рис. 5.

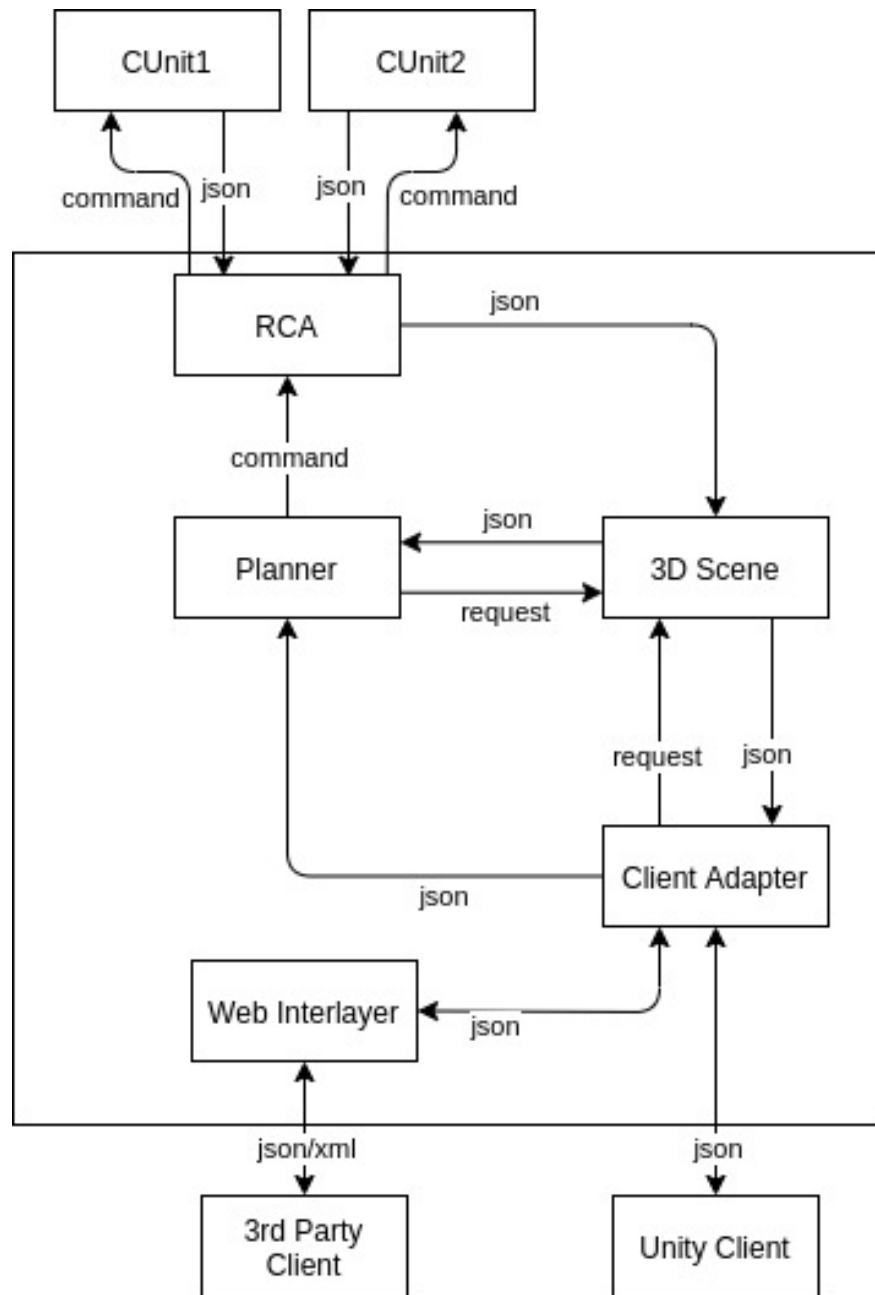


Рис. 5: Архитектура

Unity Client — это приложение, продемонстрированное в главе 3. Оно предоставляет визуальный интерфейс для управления системой роботов, может отправлять команды в Client Adapter и получать от него информацию о текущем состоянии сцены (типы и идентификаторы роботов, их

положение в выбранной системе координат, показания датчиков и положение иных объектов).

3rd Party Client — это сторонний клиент, не имеющий визуального интерфейса. С его помощью можно отправлять HTTP-запросы на выполнение некоторых команд или получение информации о состоянии сцены в формате JSON или XML.

Web Interlayer служит для приема запросов от сторонних клиентов и выполняет их первичную обработку, такую как авторизация, валидация входных данных и перевод запросов из формата XML в формат JSON. Затем этот модуль отправляет команды в Client Adapter.

Client Adapter принимает запросы от клиентов, отвечает на них, а также в зависимости от их типа отправляет новые запросы либо к модулю 3D Scene, либо к модулю Planner.

3D Scene хранит, обновляет и возвращает по запросу от других модулей информацию о текущем состоянии сцены. Он передает информацию в Client Adapter и Planner. Обновление информации в самом 3D Scene осуществляет модуль RCA с помощью специальных запросов.

Planner — модуль, осуществляющий планирование группового взаимодействия роботов на основе полученной команды среднего уровня и ретранслирующий команды в язык нижнего уровня для передачи в RCA.

RCA (Robot Control Adapter) — модуль, предоставляющий унифицированный интерфейс для взаимодействия с различными манипуляторами и различными мобильными роботами. Принимает команды от Planner, отправляет их в соответствующие модули CUnit и обновляет информацию в 3D Scene.

CUnit (Control Unit) — модуль, непосредственно взаимодействующий с конечным исполнителем в виде робота и набором датчиков, установленных на нем.

RCA, Planner, 3D Scene и Client Adapter составляют единый удаленный центр управления экспериментальным робототехническим комплексом (Remote Control Center), который позволяет осуществлять интеграцию с роботами различных производителей по унифицированному API с помощью интеграционной компоненты Control Unit.

## Глава 3. Программная реализация

### 3.1 Общее описание

При помощи 3D-редакторов и игрового движка Unity 3D была составлена модель рабочей сцены, представленной на рис. 6.

Ввиду использования Unity 3D, описанная выше математическая модель манипулятора была реализована на языке *C#*. Ссылки на ее исходный код и на весь репозиторий представлены в приложении. В архитектуру была заложена возможность на основе одного базового класса, описывающего абстрактную математическую модель робота, создавать модели конкретных роботов, используя механизм наследования. Для создания класса, описывающего модель нового робота, необходимо задать только параметры Денавита-Хартенберга, функцию перевода углов  $q$  и функцию вычисления центра сферического запястья. Основная же логика и вспомогательные методы, наподобие вычисления матрицы поворота из трех углов Эйлера или вычисления шести декартовых координат положения робота из матрицы, находятся в абстрактном базовом классе. UML-диаграмма классов математических моделей роботов представлена на рис. 7.

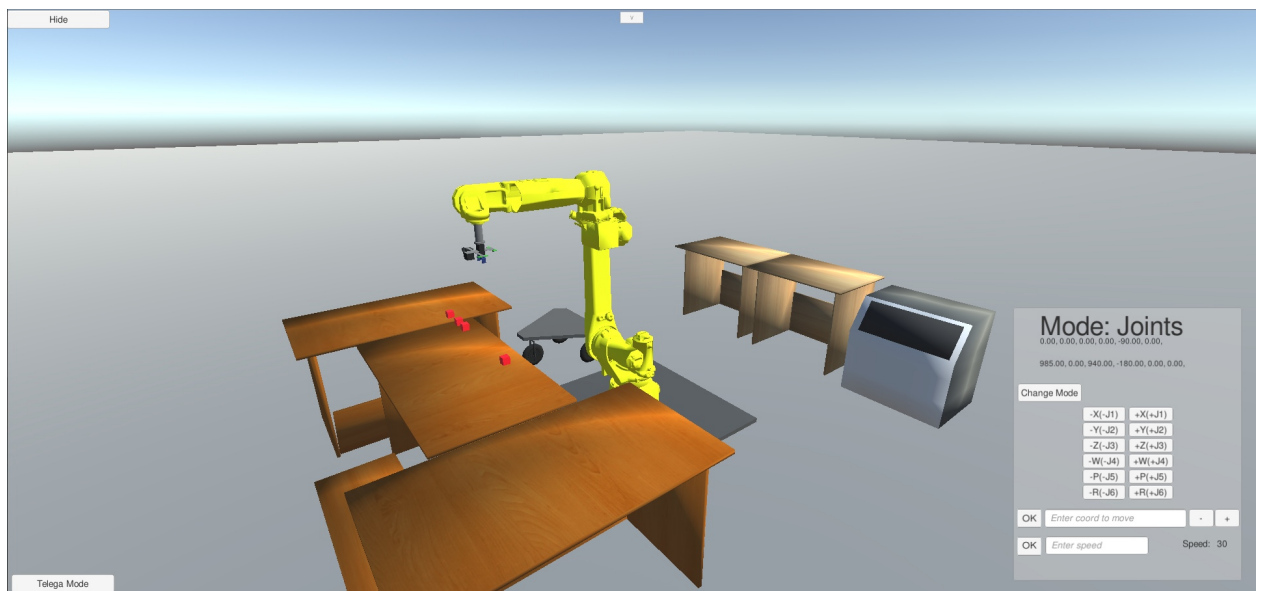


Рис. 6: Сцена

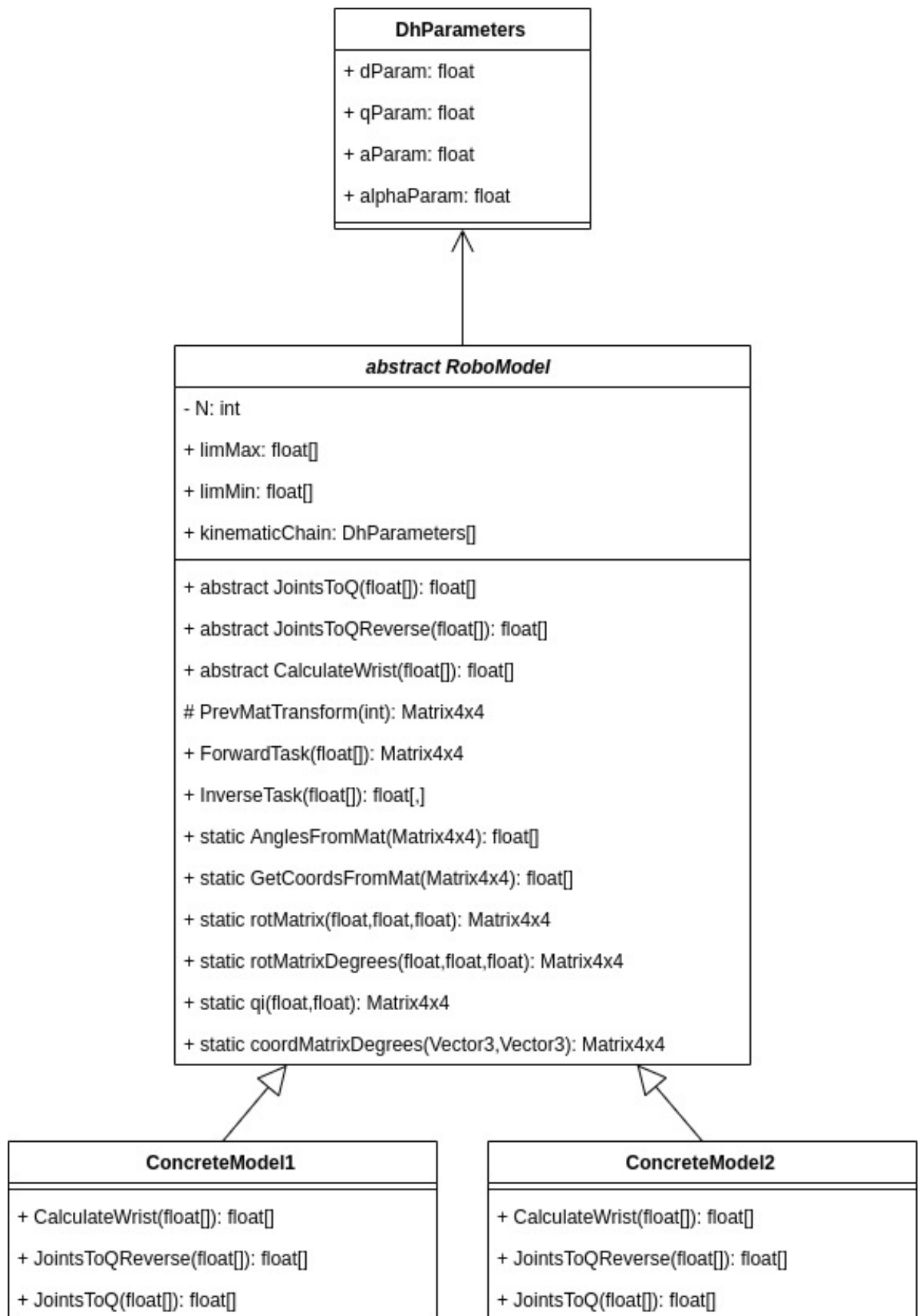


Рис. 7: Диаграмма классов

## 3.2 Формат команд для управления

Режим управления реализован на основе TCP-сокетов. По ним передаются сообщения с командами в формате JSON, которые обрабатываются другими компонентами системы. Примеры простейших команд представлены на рис. 8, 9 и 10. Сообщение со значением "0" в поле "flag" означает, что команду необходимо поставить в очередь на выполнение в модуле Planner. Внутри поля Scenario можно группировать сразу несколько команд, например, одновременно переместить колесного робота и робота-манипулятора. Таким сценариям можно задавать имена, указав значение в поле "name", чтобы была возможность их сохранить и вызывать в дальнейшем по имени в качестве идентификатора. Назначение полей в простейшей команде внутри сценария:

1. parallel: флаг, указывающий выполнять команды параллельно или последовательно,
2. name: имя робота. "f" — робот-манипулятор, "t" — колесный робот,
3. time и energy: ожидаемые время и энергия на выполнение определенной команды. Опциональные параметры, используются для построения оптимальной последовательности выполнения команд в модуле Planner,
4. command: непосредственно команда, которая должна быть выполнена роботом. Если команда начинается с символа "m", значит, это команда на перемещение и после нее должны следовать координаты. Для колесного робота — две декартовы координаты в миллиметрах и еще одно число, которое показывает тип движения, для робота-манипулятора — шесть обобщенных координат в градусах и последнее число, отвечающее за управление схватом. Например, 0 — переместиться и ничего не делать со схватом, 2 — сначала сжать схват, потом переместиться, 3 — разжать схват, потом переместиться, и так далее. Если команда состоит из символа "f", значит, это команда на определение координат объектов при помощи камеры.

```

{
  "flag": "0",
  "name": "",
  "Scenario": [
    {
      "parallel": false,
      "name": "t",
      "time": 10,
      "energy": 0,
      "command": "m -1105 -1032 0"
    }
  ]
}

```

Рис. 8: Команда на передвижение колесного робота

```

{
  "flag": "0",
  "name": "",
  "Scenario": [
    {
      "parallel": false,
      "name": "f",
      "time": 0,
      "energy": 0,
      "command": "m 10 32 11 4 -92 12 0"
    }
  ]
}

```

Рис. 9: Команда на передвижение робота-манипулятора

```

{
  "flag": "0",
  "name": "",
  "Scenario": [
    {
      "parallel": false,
      "name": "f",
      "time": 3,
      "energy": 0,
      "command": "f"
    }
  ]
}

```

Рис. 10: Команда на определение координат объектов с камеры робота-манипулятора



На рис. 11 представлен пример сложной команды, которая обозначает последовательность следующих действий:

1. переместить манипулятор в некоторую точку и разжать схват,
2. определить координаты и идентификаторы объектов, попадающих в поле зрения камеры,
3. переместить манипулятор к объекту и сжать схват,
4. переместить манипулятор вместе с объектом в другую точку.

```
{
  "flag": "0",
  "name": "TakeCubeAndMove",
  "Scenario": [
    {
      "parallel": false,
      "name": "f",
      "time": 8,
      "energy": 0,
      "command": "m -906 -462 404 180 0 180 6"
    },
    {
      "parallel": false,
      "name": "f",
      "time": 3,
      "energy": 0,
      "command": "f"
    },
    {
      "parallel": false,
      "name": "f",
      "time": 15,
      "energy": 0,
      "command": "m $object0$ + 0 0 0 0 0 0 ! 5"
    },
    {
      "parallel": false,
      "name": "f",
      "time": 16,
      "energy": 0,
      "command": "m -906 -462 940 180 0 180 0"
    }
  ]
}
```

**Рис. 11:** Сложная команда

Такой формат команд позволяет осуществить как копирующее управление отдельными роботами, так и выполнение сложных последовательно-

стей действий группой из нескольких роботов посредством специальных интерфейсов. В первом случае будет необходимо с периодичностью  $\Delta t$  отправлять простые команды на перемещение, как, например, на рис. 9, где координаты будут задаваться пользователем. Во втором случае достаточно составить желаемые сложные команды и отправить на выполнение.

### 3.3 Интерфейсы для управления

Была составлена упрощенная модель пульта управления роботом-манипулятором, изображенная на рис. 12, содержащая основные элементы, такие как кнопки для управления каждым звеном по отдельности, переключение между режимами управления в обобщенных и декартовых координатах, кнопки повышения скорости и перехода в заданную точку. Также для удобства пользователя было настроено управление при помощи клавиатуры, назначение клавиш для изменения координат робота представлено в табл. 3.

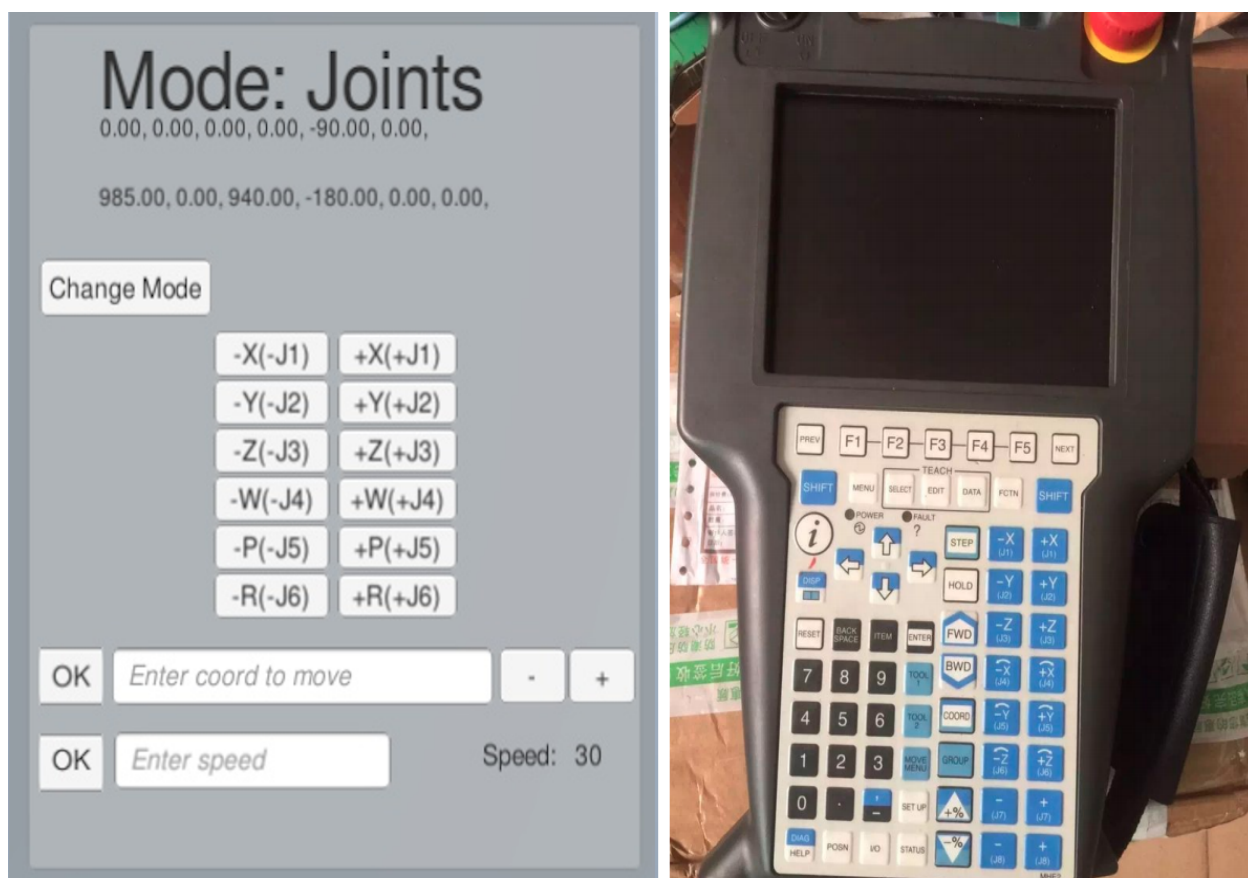
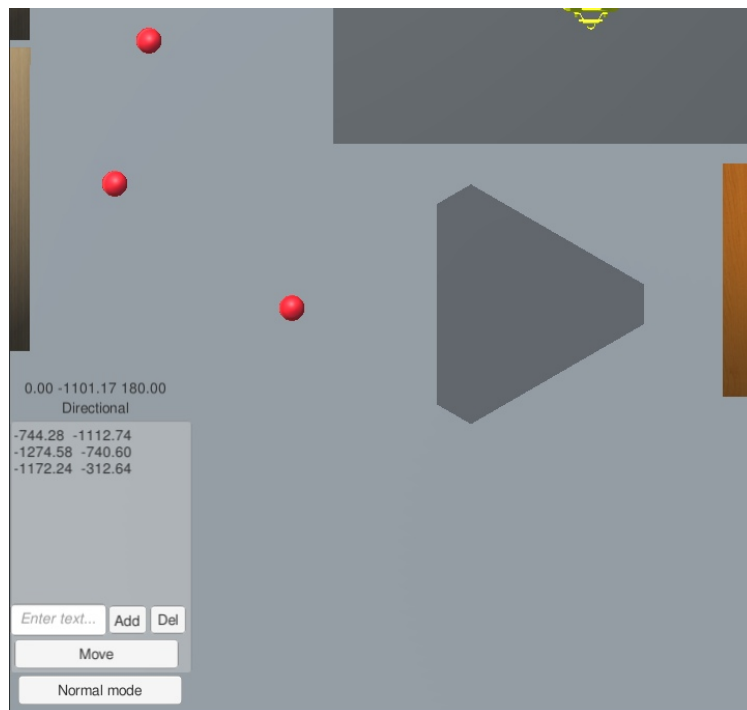


Рис. 12: Модель и реальный пульт управления

Управление колесными роботами (интерфейс изображен на рис. 13) задается упорядоченным набором контрольных точек, по которым робот последовательно проходит. Также имеется переключатель между типами движения.



**Рис. 13:** Режим управления колесным роботом

Звено/Координата	Положительное направление	Отрицательное направление
1/X	D	A
2/Y	W	S
3/Z	Z	X
4/W	E	Q
5/P	C	V
6/R	R	T

**Таблица 3:** Назначение клавиш

### 3.4 Интерфейс построения сложных команд

Для построения сложных команд был составлен специальный интерфейс. Он состоит из трех частей:

- слева находится набор доступных команд, в него входят базовые команды и сохраненные ранее сложные. На их основе составляются новые команды,
- в центре находится поле, в котором составляются новые и редактируются имеющиеся сложные команды из последовательности доступных команд, сверху можно указать имя для новой сложной команды, внизу располагаются кнопки для выбора действия с текущей командой: отправить, сохранить, переписать, выполнить в режиме симуляции,
- справа расположены окна с настройками базовых команд, они становятся доступны при нажатии кнопки Set (Settings) для соответствующей простой команды, и в них задаются параметры, необходимые для команды данного типа.

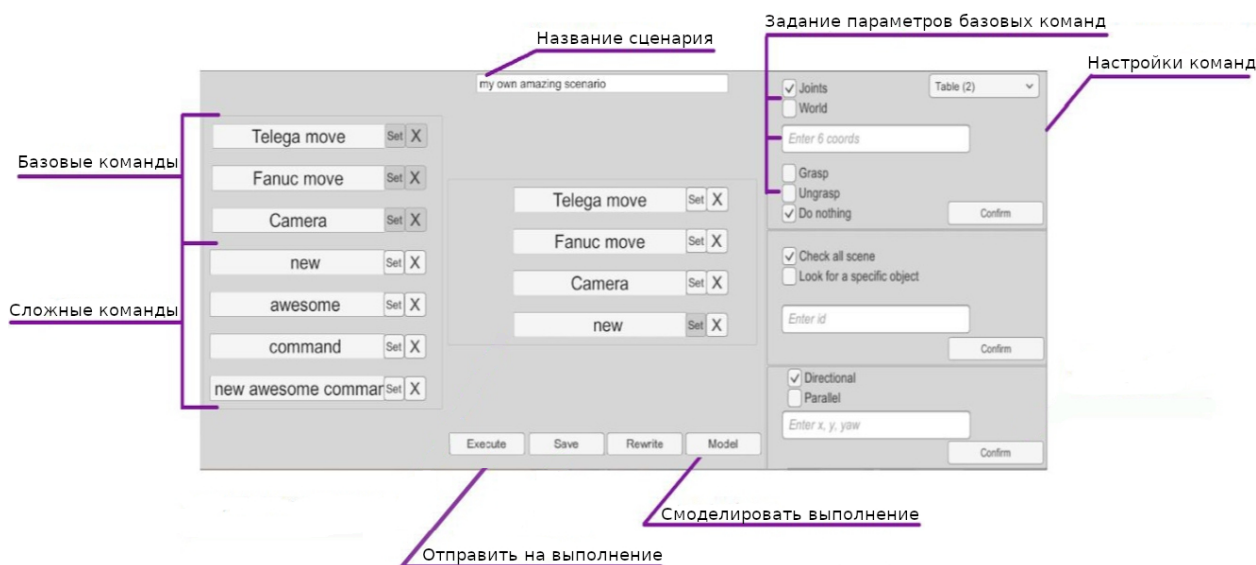


Рис. 14: Интерфейс сложных команд

### 3.5 Система технического зрения

Место крепления камеры на рабочем инструменте робота изображено на рис. 15.

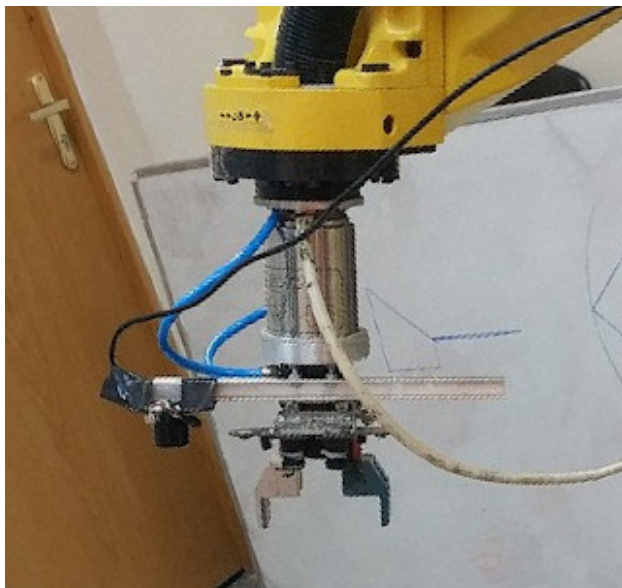


Рис. 15: Камера

Для определения положений объектов были использованы метки ArUco [6], [7]. Они позволяют идентифицировать объекты и определить их положение относительно камеры на основе изменения масштаба и искажения метки на изображении. Используемые в экспериментах объекты с метками и пример найденного положения объекта представлены на рис. 16.

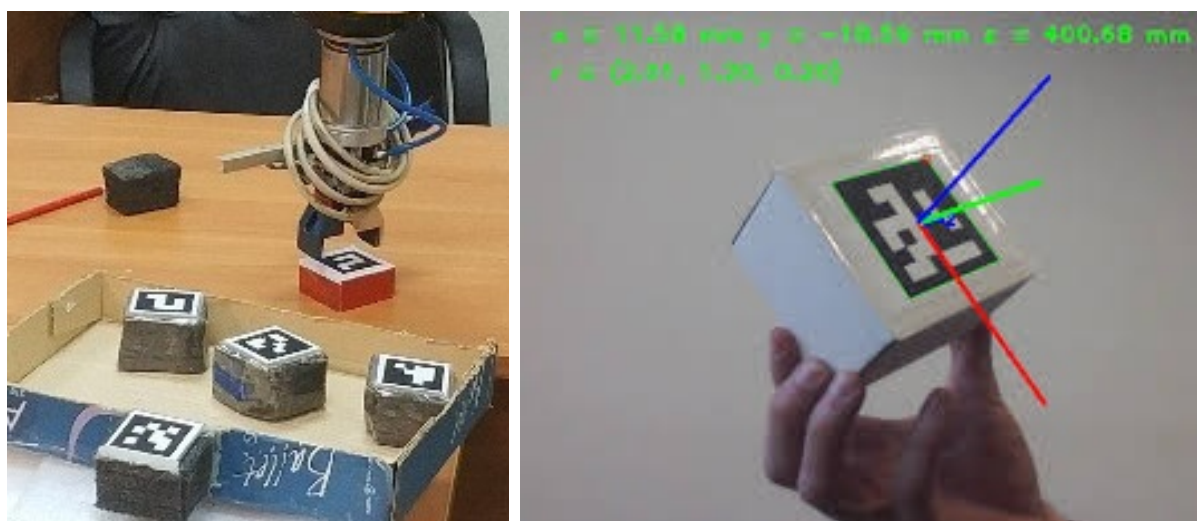


Рис. 16: Метки

## Заключение

В ходе данной работы был построен работающий прототип системы, проведены эксперименты (в приложении указана ссылка на видеозаписи с демонстрацией работы) и решены следующие задачи:

- построена компьютерная модель рабочего окружения с учетом всех измерений и расстояний. Большая часть 3D-моделей была создана собственноручно в 3D-редакторах, модель робота Fanuc была найдена в сети Интернет,
- выбраны и описаны математические модели роботов,
- реализованы математические модели роботов с применением принципов проектирования архитектуры программного обеспечения,
- обеспечено выполнение действий в режиме симуляции, копирующее управление, сохранение и исполнение последовательности команд,
- построен пользовательский интерфейс, который позволяет выполнять описанные выше операции.

## Список литературы

- [1] Горбунов В.И., Моисеев О.С., Поваляев Н.Д. Компьютерная и математическая модели мобильного трехколесного робота // Процессы управления и устойчивость, том 4, СПб, 2017. – С. 208–212.
- [2] Angeles J. Fundamentals of Robotic Mechanical Systems: Theory, Methods, and Algorithms, New York, 3rd Edition, 2007. – P. 129–162.
- [3] С.Л. Зенкевич, А.С. Ющенко. Управление роботами, Издательство МГТУ им. Н.Э. Баумана, 2000. – С. 12–101.
- [4] Moiseev, O.S., Sarsadskikh, A.S., Povalyaev, N.D., Gorbunov, V.I., Kulakov, F.M., Vasilev, V.V. Force-sensed interface for control and training space robot, 2018. // The Eighth Polyakhov's Reading: Proceedings of the International Scientific Conference on Mechanics. doi: 10.1063/1.5034735
- [5] Алимова А.В. Кинематика манипулятора «Fanuc» // Санкт-Петербургский государственный университет, 2012. – С. 6–21.
- [6] S. Garrido-Jurado, R. Muñoz-Salinas, F.J. Madrid-Cuevas, R. Medina-Carnicer, Generation of fiducial marker dictionaries using Mixed Integer Linear Programming // Pattern Recognition, Volume 51, 2016, – P. 481–491, ISSN 0031-3203, doi: 10.1016/j.patcog.2015.09.023
- [7] Francisco J. Romero-Ramirez, R. Muñoz-Salinas, R. Medina-Carnicer, Speeded up detection of squared fiducial markers // Image and Vision Computing, Volume 76, 2018, – P. 38–47, ISSN 0262-8856, doi: 10.1016/j.imavis.2018.05.004

## Приложение

1. Ссылка на репозиторий:

<https://github.com/nikita03565/FanucInterface>

2. Абстрактный класс математической модели шестизвеного робота манипулятора:

<https://github.com/nikita03565/FanucInterface/blob/master/Assets/Scripts/Fanuc/RoboModel.cs>

3. Класс математической модели робота Fanuc M-20iA:

<https://github.com/nikita03565/FanucInterface/blob/master/Assets/Scripts/Fanuc/FanucModel.cs>

4. Ссылка на папку в Google Drive с видеозаписями работы системы:

<https://drive.google.com/drive/u/0/folders/1jpDoYgMIE8VaQN7xJxSTqFfseJqBesS9>